Министерство образования и науки Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение науки и высшего образования**

«Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра информационных систем и программной инженерии

**Лабораторная работа № 5**

**по дисциплине**

**«Программирование компьютерной графики»**

**Тема: «Инициализация OpenGL»**

Выполнил:

ст. гр. ПРИ-117

Хлызова В.Г.

Принял:

Жигалов И.Е.

Владимир, 2020

ЦЕЛЬ РАБОТЫ

Изучение принципов инициализация OpenGL и визуализация объектов при разработке приложений в C#.

ЗАДАНИЕ

1. Ознакомиться по методическим указаниям и литературе с теоретическим материалом.

2. Выполнить действия, приведенные в разделах 5.2, 5.3, 5.4. При разработке программы имя проекта, создаваемого в MS Visual Studio, должно содержать фамилию студента и группу (например, Ivanov\_Ivan\_ISG\_105\_lab\_1).

3. При выполнении задания для отрисовки линий использовать цвет, указанный в варианте задания.

4. При рисовании устанавливать толщину линий (функция glLineWidth) из варианта задания.

5. При выполнении задания раздела 5.2 визуализировать с помощью линий свою фамилию и инициалы.

6. При выполнении задания раздела 5.3 использовать функцию, указанную в варианте.

7. При выполнении задания раздела 5.4 добавить в окно и в его заголовок свою фамилию, имя и группу.

ХОД РАБОТЫ

Визуализация 2D примитивов

using System;

using System.Windows.Forms;

using Tao.OpenGl;

using Tao.FreeGlut;

namespace Khlyzova\_Valeria\_PRI\_117\_lab\_05

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

AnT.InitializeContexts();

}

private void Form1\_Load(object sender, EventArgs e)

{

Glut.glutInit();

Glut.glutInitDisplayMode(Glut.GLUT\_RGB | Glut.GLUT\_DOUBLE | Glut.GLUT\_DEPTH);

Gl.glClearColor(255, 255, 255, 1);

Gl.glViewport(0, 0, AnT.Width, AnT.Height);

Gl.glMatrixMode(Gl.GL\_PROJECTION);

Gl.glLoadIdentity();

if ((float)AnT.Width <= (float)AnT.Height)

{

Glu.gluOrtho2D(0.0, 30.0 \* (float)AnT.Height / (float)AnT.Width, 0.0, 30.0);

}

else

{

Glu.gluOrtho2D(0.0, 30.0 \* (float)AnT.Width / (float)AnT.Height, 0.0, 30.0);

}

Gl.glMatrixMode(Gl.GL\_MODELVIEW);

Gl.glLoadIdentity();

Gl.glEnable(Gl.GL\_DEPTH\_TEST);

Gl.glEnable(Gl.GL\_COLOR\_MATERIAL);

}

private void button1\_Click(object sender, EventArgs e)

{

Gl.glClear(Gl.GL\_COLOR\_BUFFER\_BIT | Gl.GL\_DEPTH\_BUFFER\_BIT);

Gl.glLoadIdentity();

Gl.glColor3f(255, 0, 0);

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(8, 7);

Gl.glVertex2d(15, 27);

Gl.glVertex2d(17, 27);

Gl.glVertex2d(23, 7);

Gl.glVertex2d(21, 7);

Gl.glVertex2d(19, 14);

Gl.glVertex2d(12.5, 14);

Gl.glVertex2d(10, 7);

Gl.glEnd();

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(18.5, 16);

Gl.glVertex2d(16, 25);

Gl.glVertex2d(13.2, 16);

Gl.glEnd();

Gl.glFlush();

AnT.Invalidate();

}

private void button2\_Click(object sender, EventArgs e)

{

Gl.glClear(Gl.GL\_COLOR\_BUFFER\_BIT | Gl.GL\_DEPTH\_BUFFER\_BIT);

Gl.glLoadIdentity();

Gl.glLineWidth(2);

// X

Gl.glColor3f(0, 0, 255);

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(1, 13);

Gl.glVertex2d(3, 19);

Gl.glVertex2d(1, 25);

Gl.glVertex2d(3, 25);

Gl.glVertex2d(5, 20);

Gl.glVertex2d(7, 25);

Gl.glVertex2d(9, 25);

Gl.glVertex2d(7, 19);

Gl.glVertex2d(9, 13);

Gl.glVertex2d(7, 13);

Gl.glVertex2d(5, 18);

Gl.glVertex2d(3, 13);

Gl.glEnd();

// Л

Gl.glColor3f(0, 255, 0);

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(7, 13);

Gl.glVertex2d(11, 25);

Gl.glVertex2d(13, 25);

Gl.glVertex2d(17, 13);

Gl.glVertex2d(15, 13);

Gl.glVertex2d(12, 23);

Gl.glVertex2d(9, 13);

Gl.glEnd();

// Ы

Gl.glColor3f(255, 0, 0);

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(15, 13);

Gl.glVertex2d(23, 13);

Gl.glVertex2d(23, 19);

Gl.glVertex2d(17, 19);

Gl.glVertex2d(17, 25);

Gl.glVertex2d(15, 25);

Gl.glEnd();

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(17, 15);

Gl.glVertex2d(21, 15);

Gl.glVertex2d(21, 17);

Gl.glVertex2d(17, 17);

Gl.glEnd();

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(24, 13);

Gl.glVertex2d(26, 13);

Gl.glVertex2d(26, 25);

Gl.glVertex2d(24, 25);

Gl.glEnd();

// З

Gl.glColor3f(255, 255, 0);

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(25, 13);

Gl.glVertex2d(33, 13);

Gl.glVertex2d(33, 25);

Gl.glVertex2d(25, 25);

Gl.glVertex2d(25, 23);

Gl.glVertex2d(31, 23);

Gl.glVertex2d(31, 20);

Gl.glVertex2d(27, 20);

Gl.glVertex2d(27, 18);

Gl.glVertex2d(31, 18);

Gl.glVertex2d(31, 15);

Gl.glVertex2d(25, 15);

Gl.glEnd();

// О

Gl.glColor3f(255, 0, 255);

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(32, 13);

Gl.glVertex2d(41, 13);

Gl.glVertex2d(41, 25);

Gl.glVertex2d(32, 25);

Gl.glEnd();

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(34, 15);

Gl.glVertex2d(39, 15);

Gl.glVertex2d(39, 23);

Gl.glVertex2d(34, 23);

Gl.glEnd();

// В

Gl.glColor3f(0, 255, 255);

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(40, 13);

Gl.glVertex2d(40, 25);

Gl.glVertex2d(45, 25);

Gl.glVertex2d(45, 20);

Gl.glVertex2d(50, 20);

Gl.glVertex2d(50, 13);

Gl.glEnd();

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(41, 15);

Gl.glVertex2d(41, 18);

Gl.glVertex2d(48, 18);

Gl.glVertex2d(48, 15);

Gl.glEnd();

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(41, 23);

Gl.glVertex2d(41, 21);

Gl.glVertex2d(44, 21);

Gl.glVertex2d(44, 23);

Gl.glEnd();

// А

Gl.glColor3f(0, 0, 255);

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(49, 13);

Gl.glVertex2d(53, 25);

Gl.glVertex2d(55, 25);

Gl.glVertex2d(59, 13);

Gl.glVertex2d(57, 13);

Gl.glVertex2d(56, 17);

Gl.glVertex2d(52, 17);

Gl.glVertex2d(51, 13);

Gl.glEnd();

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(55, 19);

Gl.glVertex2d(53, 19);

Gl.glVertex2d(54, 23);

Gl.glEnd();

// В - Имя Валерия

Gl.glColor3f(255, 0, 0);

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(40, 0);

Gl.glVertex2d(40, 12);

Gl.glVertex2d(45, 12);

Gl.glVertex2d(45, 7);

Gl.glVertex2d(50, 7);

Gl.glVertex2d(50, 0);

Gl.glEnd();

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(41, 2);

Gl.glVertex2d(41, 5);

Gl.glVertex2d(48, 5);

Gl.glVertex2d(48, 2);

Gl.glEnd();

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(41, 10);

Gl.glVertex2d(41, 8);

Gl.glVertex2d(44, 8);

Gl.glVertex2d(44, 10);

Gl.glEnd();

Gl.glFlush();

// Г - Отчество Григорьевна

Gl.glColor3f(0, 0, 255);

Gl.glBegin(Gl.GL\_LINE\_LOOP);

Gl.glVertex2d(49, 0);

Gl.glVertex2d(49, 12);

Gl.glVertex2d(57, 12);

Gl.glVertex2d(57, 10);

Gl.glVertex2d(51, 10);

Gl.glVertex2d(51, 0);

Gl.glEnd();

Gl.glFlush();

AnT.Invalidate();

}

private void button3\_Click(object sender, EventArgs e)

{

Form2 f2 = new Form2();

f2.Show();

}

private void button4\_Click(object sender, EventArgs e)

{

Form3 f3 = new Form3();

f3.Show();

}

}

}

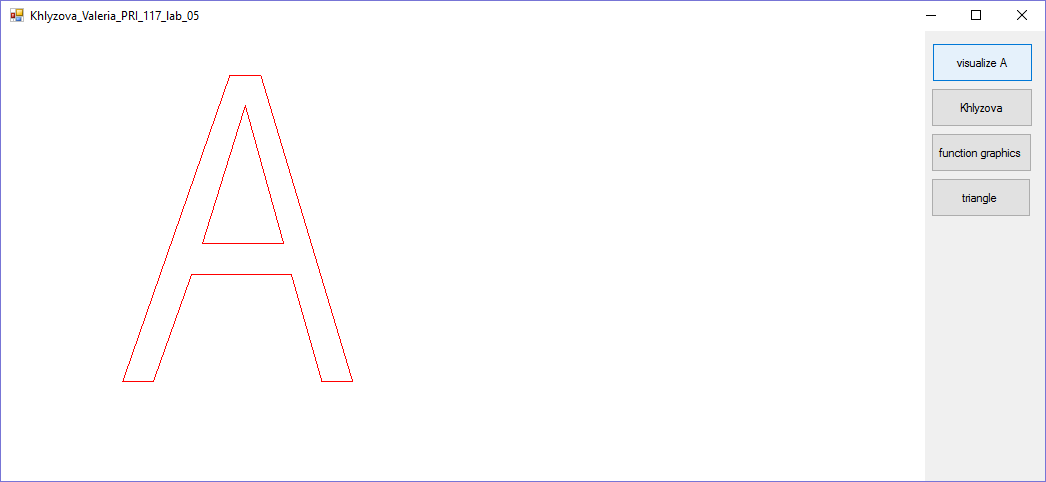


Рисунок 1. Визуализация буквы А.

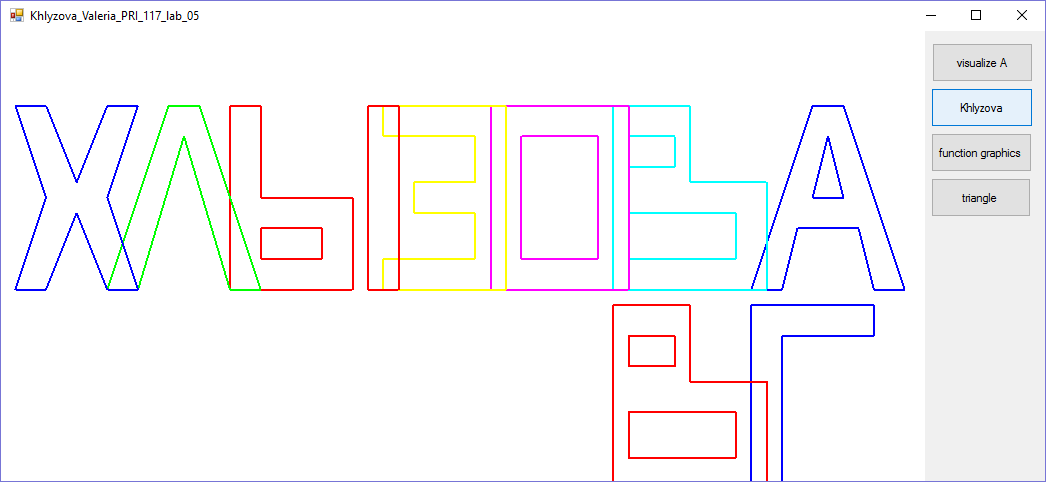


Рисунок 2. Визуализация фамилии и инициалов.

Визуализация графика функции

Вариант 15

using System;

using System.Windows.Forms;

using Tao.OpenGl;

using Tao.FreeGlut;

namespace Khlyzova\_Valeria\_PRI\_117\_lab\_05

{

public partial class Form2 : Form

{

double ScreenW, ScreenH;

private float devX;

private float devY;

private float[,] GrapValuesArray;

private int elements\_count = 0;

private bool not\_calculate = true;

private int pointPosition = 0;

float lineX, lineY;

float Mcoord\_X = 0, Mcoord\_Y = 0;

public Form2()

{

InitializeComponent();

AnT.InitializeContexts();

}

private void Form2\_Load(object sender, EventArgs e)

{

Glut.glutInitDisplayMode(Glut.GLUT\_RGB | Glut.GLUT\_DOUBLE);

Gl.glClearColor(255, 255, 255, 1);

Gl.glViewport(0, 0, AnT.Width, AnT.Height);

Gl.glMatrixMode(Gl.GL\_PROJECTION);

Gl.glLoadIdentity();

if ((float)AnT.Width <= (float)AnT.Height)

{

ScreenW = 30.0;

ScreenH = 30.0 \* (float)AnT.Height / (float)AnT.Width;

Glu.gluOrtho2D(0.0, ScreenW, 0.0, ScreenH);

}

else

{

ScreenW = 30.0 \* (float)AnT.Width / (float)AnT.Height;

ScreenH = 30.0;

Glu.gluOrtho2D(0.0, 30.0 \* (float)AnT.Width / (float)AnT.Height, 0.0, 30.0);

}

devX = (float)ScreenW / (float)AnT.Width;

devY = (float)ScreenH / (float)AnT.Height;

Gl.glMatrixMode(Gl.GL\_MODELVIEW);

PointInGrap.Start();

}

private void PointInGrap\_Tick(object sender, EventArgs e)

{

if (pointPosition == elements\_count - 1) pointPosition = 0;

Draw();

pointPosition++;

}

private void AnT\_MouseMove(object sender, MouseEventArgs e)

{

Mcoord\_X = e.X;

Mcoord\_Y = e.Y;

lineX = devX \* e.X;

lineY = (float)(ScreenH - devY \* e.Y);

}

private void PrintText2D(float x, float y, string text)

{

Gl.glRasterPos2f(x, y);

foreach (char char\_for\_draw in text)

{

Glut.glutBitmapCharacter(Glut.GLUT\_BITMAP\_9\_BY\_15, char\_for\_draw);

}

}

private void functionCalculation()

{

float x = 0, y = 0;

float a = -5, b = 5, n = 40;

GrapValuesArray = new float[300, 2];

elements\_count = 0;

for (x = a; x < b; x += (b - a)/n)

{

// 7 \* x / (3 \* x ^ 2 + 2 \* x + 1)

y = (float) 7 \* x / (3 \* (float)Math.Pow(x, 2) + 2 \* x + 1);

GrapValuesArray[elements\_count, 0] = x;

GrapValuesArray[elements\_count, 1] = y;

elements\_count++;

}

not\_calculate = false;

}

private void DrawDiagram()

{

if (not\_calculate)

{

functionCalculation();

}

Gl.glBegin(Gl.GL\_LINE\_STRIP);

Gl.glVertex2d(GrapValuesArray[0, 0], GrapValuesArray[0, 1]);

for (int ax = 1; ax < elements\_count; ax += 2)

{

Gl.glVertex2d(GrapValuesArray[ax, 0], GrapValuesArray[ax, 1]);

}

Gl.glEnd();

Gl.glPointSize(5);

Gl.glColor3f(255, 0, 0);

Gl.glBegin(Gl.GL\_POINTS);

Gl.glVertex2d(GrapValuesArray[pointPosition, 0], GrapValuesArray[pointPosition, 1]);

Gl.glEnd();

Gl.glPointSize(1);

}

private void Draw()

{

Gl.glClear(Gl.GL\_COLOR\_BUFFER\_BIT | Gl.GL\_DEPTH\_BUFFER\_BIT);

Gl.glLoadIdentity();

Gl.glColor3f(0, 0, 0);

Gl.glPushMatrix();

Gl.glTranslated(15, 15, 0);

Gl.glBegin(Gl.GL\_POINTS);

for (int ax = -15; ax < 15; ax++)

{

for (int bx = -15; bx < 15; bx++)

{

Gl.glVertex2d(ax, bx);

}

}

Gl.glEnd();

Gl.glBegin(Gl.GL\_LINES);

Gl.glVertex2d(0, -15);

Gl.glVertex2d(0, 15);

Gl.glVertex2d(-15, 0);

Gl.glVertex2d(15, 0);

Gl.glVertex2d(0, 15);

Gl.glVertex2d(0.1, 14.5);

Gl.glVertex2d(0, 15);

Gl.glVertex2d(-0.1, 14.5);

Gl.glVertex2d(15, 0);

Gl.glVertex2d(14.5, 0.1);

Gl.glVertex2d(15, 0);

Gl.glVertex2d(14.5, -0.1);

Gl.glEnd();

PrintText2D(15.5f, 0, "x");

PrintText2D(0.5f, 14.5f, "y");

DrawDiagram();

Gl.glPopMatrix();

PrintText2D(devX \* Mcoord\_X + 0.2f, (float)ScreenH - devY \* Mcoord\_Y + 0.4f, "[ x: " + (devX \* Mcoord\_X - 15).ToString() + " ; y: " + ((float)ScreenH - devY \* Mcoord\_Y - 15).ToString() + "]");

Gl.glColor3f(255, 0, 0);

Gl.glBegin(Gl.GL\_LINES);

Gl.glVertex2d(lineX, 15);

Gl.glVertex2d(lineX, lineY);

Gl.glVertex2d(15, lineY);

Gl.glVertex2d(lineX, lineY);

Gl.glEnd();

Gl.glFlush();

AnT.Invalidate();

}

}

}

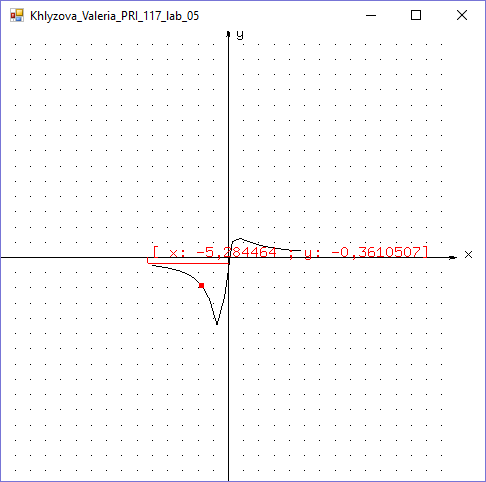


Рисунок 3. График функции по варианту.

Вывод 2D цветового треугольника

using System;

using Tao.OpenGl;

using Tao.FreeGlut;

using System.Windows.Forms;

namespace Khlyzova\_Valeria\_PRI\_117\_lab\_05

{

public partial class Form3 : Form

{

double a = 1, b = 0, c = 0;

public Form3()

{

InitializeComponent();

AnT.InitializeContexts();

}

private void Form3\_Load(object sender, EventArgs e)

{

Glut.glutInitDisplayMode(Glut.GLUT\_RGB | Glut.GLUT\_DOUBLE | Glut.GLUT\_DEPTH);

Gl.glClearColor(255, 255, 255, 1);

Gl.glViewport(0, 0, AnT.Width, AnT.Height);

Gl.glMatrixMode(Gl.GL\_PROJECTION);

Gl.glLoadIdentity();

if (AnT.Width <= AnT.Height)

Glu.gluOrtho2D(0.0, 30.0, 0.0, 30.0 \* (float)AnT.Height / (float)AnT.Width);

else

Glu.gluOrtho2D(0.0, 30.0 \* (float)AnT.Width / (float)AnT.Height, 0.0, 30.0);

Gl.glMatrixMode(Gl.GL\_MODELVIEW);

}

private void trackBar1\_Scroll(object sender, EventArgs e)

{

a = (double)trackBar1.Value / 1000.0;

label4.Text = a.ToString();

}

private void trackBar2\_Scroll(object sender, EventArgs e)

{

b = (double)trackBar2.Value / 1000.0;

label5.Text = b.ToString();

}

private void trackBar3\_Scroll(object sender, EventArgs e)

{

c = (double)trackBar3.Value / 1000.0;

label6.Text = c.ToString();

}

private void label4\_Click(object sender, EventArgs e)

{

}

private void timer1\_Tick(object sender, EventArgs e)

{

Draw();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void button1\_Click\_1(object sender, EventArgs e)

{

RenderTimer.Start();

}

private void Draw()

{

Gl.glClear(Gl.GL\_COLOR\_BUFFER\_BIT);

Gl.glBegin(Gl.GL\_TRIANGLES);

Gl.glColor3d(0.0 + a, 0.0 + b, 0.0 + c);

Gl.glVertex2d(5.0, 5.0);

Gl.glColor3d(0.0 + c, 0.0 + a, 0.0 + b);

Gl.glVertex2d(25.0, 5.0);

Gl.glColor3d(0.0 + b, 0.0 + c, 0.0 + a);

Gl.glVertex2d(5.0, 25.0);

Gl.glEnd();

Gl.glFlush();

AnT.Invalidate();

}

}

}

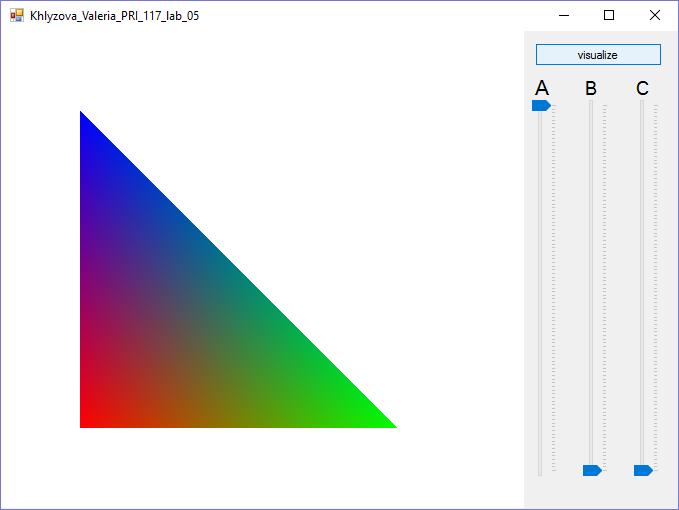


Рисунок 4. Вывод 2D цветового треугольника.

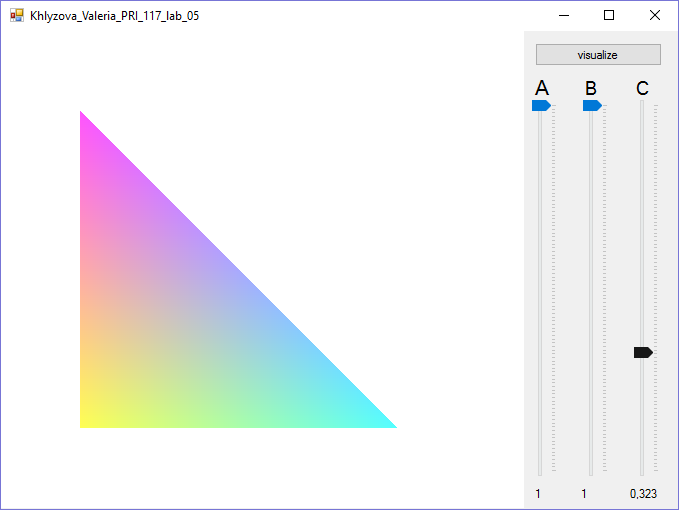


Рисунок 5. Изменение настроек.

ВЫВОД

В ходе выполнения лабораторной работы были изучены принципы инициализации OpenGL и визуализации объектов при разработке приложений в C#.